This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently amended) An asynchronous messaging architecture for processing messages, comprising:

an instance of an automated business process, the automated business process including response processing code, the response processing code including exception handling code;

~~means for~~ a program manager configured to manage the instance of the automated business process;

the program manager further configured to detect ~~detecting~~ when [[an]] the instance of [[an]] the automated business process is waiting for a response to a message, wherein a response indicates a success or failure of the message;

~~means for~~ the program manager further configured to store, when the instance is waiting for the response, at least a part of state information associated with the instance in a database and remove the instance from active memory ~~storing the instance so as to suspend processing of the instance~~;

~~means for~~ the program manager further configured to determine ~~detecting~~ when the response associated with the instance has been received and the program manager further configured to restore the instance from the database into memory and pass the instance the message ~~for resuming processing of the instance~~; and

~~means for~~ the instance further configured to process ~~processing~~ the response using the response processing code and handle exceptions using the exception handling code within the instance ~~according to the success or failure of the message, wherein the processing code has failure handling functionality~~.

2. (Currently amended) The architecture of claim 1, wherein the exception handling ~~response processing~~ code is a try-catch block.

3. (Original) The architecture of claim 1, wherein storing the instance takes place after a predetermined time.

4. (Canceled)

5. (Original) The architecture of claim 1, wherein the response is received on a port defined by the instance.

6. (Original) The architecture of claim 1, wherein the response is a response indicative of whether or not the message was received by an intended recipient.

7. (Currently amended) A method for processing a message in an asynchronous architecture, comprising:

determining that a response to a message sent by an instance of software code is to be received, wherein the response indicates a success or failure of the message;

determining whether the response has been received and, if the response has not been received, storing the instance of the software code in memory, thereby suspending the instance;

receiving the response and resuming the instance; and

processing the response using response processing code within the instance according to the success or failure of the message, wherein the response processing code has failure handling functionality.

8. (Original) The method of claim 7, wherein determining that the response is to be received is determined by encountering a catch block within the instance.

9. (Original) The method of claim 8, wherein processing the response comprises determining whether the response indicates a failure and, if so, processing the response using the catch block.

10. (Original) The method of claim 9, further comprising, if the response indicates a success, processing the response by way of the instance of the software code.

11. (Original) The method of claim 7, wherein storing the instance occurs after a predetermined time.

12. (Original) The method of claim 7, wherein storing the instance comprises storing the instance in a database and removing the instance from active memory.

13. (Original) The method of claim 12, wherein resuming the instance comprises removing the instance from the database and restoring the instance to active memory.

14. (Original) The method of claim 7, wherein the response is received on a port defined by the instance.

15. (Original) The method of claim 7, wherein the asynchronous architecture is implemented by way of distributed business process automation software.

16. (Original) The method of claim 7, wherein the message is to be received by a remote computer.

17. (Currently amended) A method for processing a message in an asynchronous architecture, comprising:

encountering a catch block in an instance of running business process automation software, wherein the catch block indicates that a response to a message is to be received;

determining whether the response has been received, wherein the response indicates a success or failure of the message, and if the response has not been received, storing the instance of the software code in memory, thereby suspending the instance;

receiving the response and resuming the instance in accordance with the receipt of the response; and

processing the response using response processing code within the instance according to the success or failure of the message, wherein the response processing code has failure handling functionality.

18. (Original) The method of claim 17, wherein processing the response comprises determining whether the response indicates a success or failure of the message and, if the response indicates a failure, processing the response using the catch block.

19. (Original) The method of claim 18, further comprising, if the response is indicative of a success, processing the response within the instance of the automation software and logically after the catch block.

20. (Original) The method of claim 17, wherein the response is received on a port defined by the instance.

21. (Currently amended) A computer-readable storage medium having computer-readable instructions for performing a method for processing a message in an asynchronous architecture, the method comprising:

determining that a response to a message sent by an instance of software code is to be received, wherein the response indicates a success or failure of the message;

determining whether the response has been received and, if the response has not been received, storing the instance of the software code in memory, thereby suspending the instance;

receiving the response and resuming the instance; and

processing the response using response processing code within the instance according to the success or failure of the message, wherein the response processing code has failure handling functionality.

22. (Original) The computer-readable medium of claim 21, wherein determining that the response is to be received is determined by encountering a catch block within the instance.

23. (Original) The computer-readable medium of claim 22, wherein processing the response comprises determining whether the response indicates a failure and, if so, processing the response using the catch block.

24. (Original) The computer-readable medium of claim 23, further comprising, if the response indicates a success, processing the response by way of the instance of the software code.

25. (Original) The computer-readable medium of claim 21, wherein storing the instance occurs after a predetermined time.

26. (Original) The computer-readable medium of claim 21, wherein storing the instance comprises storing the instance in a database and removing the instance from active memory.

27. (Original) The computer-readable medium of claim 26, wherein resuming the instance comprises removing the instance from the database and restoring the instance to active memory.

28. (Original) The computer-readable medium of claim 21, wherein the response is received on a port defined by the instance.

29. (Original) The computer-readable medium of claim 21, wherein the asynchronous architecture is implemented by way of distributed business process automation software.

30. (Original) The computer-readable medium of claim 21, wherein the message is to be received by a remote computer.

31. (Currently amended) A computer-readable medium having computer-readable instructions for performing a method for processing a message in an asynchronous architecture, the method comprising:

encountering a catch block in an instance of running business process automation software, wherein the catch block indicates that a response to a message is to be received;

determining whether the response has been received, wherein the response indicates a success or failure of the message and, if the response has not been received, storing the instance of the software code in memory, thereby suspending the instance;

receiving the response and resuming the instance in accordance with the receipt of the response; and

processing the response using response processing code within the instance according to the success or failure of the message, wherein the response processing code has failure handling functionality.

32. (Original) The computer-readable medium of claim 31, wherein processing the response comprises determining whether the response indicates a success or failure of the message and, if the response indicates a failure, processing the response using the catch block.

33. (Original) The computer-readable medium of claim 32, further comprising, if the response is indicative of a success, processing the response within the instance of the automation software and logically after the catch block.

34. (Original) The computer-readable medium of claim 31, wherein the response is received on a port defined by the instance.